

# Package ‘worldmet’

February 6, 2023

**Type** Package

**Title** Import Surface Meteorological Data from NOAA Integrated Surface Database (ISD)

**Version** 0.9.7

**Date** 2023-02-01

**ByteCompile** true

**Depends** R (>= 3.2.0)

**Imports** openair, doParallel, parallel, foreach, purrr (>= 1.0.0),  
dplyr, leaflet, tidyr, readr, magrittr, tibble

**Maintainer** David Carslaw <david.carslaw@york.ac.uk>

**Description** Functions to import data from more than 30,000 surface meteorological sites around the world managed by the National Oceanic and Atmospheric Administration (NOAA) Integrated Surface Database.

**License** GPL (>= 2)

**URL** <https://davidcarslaw.github.io/worldmet/index.html>,  
<http://davidcarslaw.github.io/worldmet/>

**BugReports** <https://github.com/davidcarslaw/worldmet/issues>

**Suggests** knitr, rmarkdown

**Language** en-GB

**LazyLoad** true

**LazyData** true

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** David Carslaw [aut, cre]

**Repository** CRAN

**Date/Publication** 2023-02-06 09:40:02 UTC

## R topics documented:

exportADMS . . . . .	2
getMeta . . . . .	3
getMetaLive . . . . .	4
importNOAA . . . . .	5
weatherCodes . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

exportADMS	<i>Export a meteorological data frame in ADMS format</i>
------------	--

---

### Description

Writes a text file in the ADMS format to a location of the user's choosing, with optional interpolation of missing values.

### Usage

```
exportADMS(dat, out = "./ADMS_met.MET", interp = FALSE, maxgap = 2)
```

### Arguments

dat	A data frame imported by <code>importNOAA()</code> .
out	A file name for the ADMS file. The file is written to the working directory by default.
interp	Should interpolation of missing values be undertaken? If TRUE linear interpolation is carried out for gaps of up to and including maxgap.
maxgap	The maximum gap in hours that should be interpolated where there are missing data when <code>interp = TRUE</code> . Data with gaps more than maxgap are left as missing.

### Value

`exportADMS()` returns the input `dat` invisibly.

### Examples

```
## Not run:
## import some data then export it
dat <- importNOAA(year = 2012)
exportADMS(dat, out = "~/adms_met.MET")

## End(Not run)
```

---

`getMeta`*Find a ISD site code and other meta data*

---

## Description

Get information on meteorological sites

## Usage

```
getMeta(  
  site = "heathrow",  
  lat = NA,  
  lon = NA,  
  country = NA,  
  state = NA,  
  n = 10,  
  end.year = "current",  
  provider = c("OpenStreetMap", "Esri.WorldImagery"),  
  plot = TRUE,  
  returnMap = FALSE  
)
```

## Arguments

<code>site</code>	A site name search string e.g. <code>site = "heathrow"</code> . The search strings and be partial and can be upper or lower case e.g. <code>site = "HEATHR"</code> .
<code>lat</code>	A latitude in decimal degrees to search. Takes the values -90 to 90.
<code>lon</code>	A longitude in decimal degrees to search. Takes values -180 to 180. Negative numbers are west of the Greenwich meridian.
<code>country</code>	The country code. This is a two letter code. For a full listing see <a href="https://www1.ncdc.noaa.gov/pub/data/noaa/isd-history.csv">https://www1.ncdc.noaa.gov/pub/data/noaa/isd-history.csv</a> .
<code>state</code>	The state code. This is a two letter code.
<code>n</code>	The number of nearest sites to search based on latitude and longitude.
<code>end.year</code>	To help filter sites based on how recent the available data are. <code>end.year</code> can be "current", "any" or a numeric year such as 2016, or a range of years e.g. 1990:2016 (which would select any site that had an end date in that range. <b>By default only sites that have some data for the current year are returned.</b>
<code>provider</code>	By default a map will be created in which readers may toggle between a vector base map and a satellite/aerial image. <code>provider</code> allows users to override this default; see <a href="http://leaflet-extras.github.io/leaflet-providers/preview/">http://leaflet-extras.github.io/leaflet-providers/preview/</a> for a list of all base maps that can be used. If multiple base maps are provided, they can be toggled between using a "layer control" interface.
<code>plot</code>	If TRUE will plot sites on an interactive leaflet map.
<code>returnMap</code>	Should the leaflet map be returned instead of the meta data? Default is FALSE.

## Details

This function is primarily used to find a site code that can be used to access data using `importNOAA()`. Sites searches of approximately 30,000 sites can be carried out based on the site name and based on the nearest locations based on user-supplied latitude and longitude.

## Value

A data frame is returned with all available meta data, mostly importantly including a code that can be supplied to `importNOAA()`. If latitude and longitude searches are made an approximate distance, `dist` in km is also returned.

## Author(s)

David Carslaw

## See Also

`getMetaLive()` to download the all meta data to allow re-use and direct querying.

## Examples

```
## Not run:
## search for sites with name beijing
getMeta(site = "beijing")

## End(Not run)

## Not run:
## search for near a specified lat/lon - near Beijing airport
## returns 'n' nearest by default
getMeta(lat = 40, lon = 116.9)

## End(Not run)
```

---

getMetaLive

*Obtain site meta data from NOAA server*

---

## Description

Download all NOAA meta data, allowing for re-use and direct querying.

## Usage

```
getMetaLive(...)
```

## Arguments

... Currently unused.

**Value**

a [tibble](#)

**Examples**

```
## Not run:
meta <- getMetaLive()
head(meta)

## End(Not run)
```

---

importNOAA	<i>Import Meteorological data from the NOAA Integrated Surface Database (ISD)</i>
------------	---

---

**Description**

This is the main function to import data from the NOAA Integrated Surface Database (ISD). The ISD contains detailed surface meteorological data from around the world for over 30,000 locations. For general information of the ISD see [website](#).

**Usage**

```
importNOAA(
  code = "037720-99999",
  year = 2014,
  hourly = TRUE,
  n.cores = 1,
  quiet = FALSE,
  path = NA
)
```

**Arguments**

code	The identifying code as a character string. The code is a combination of the USAF and the WBAN unique identifiers. The codes are separated by a "-" e.g. code = "037720-99999".
year	The year to import. This can be a vector of years e.g. year = 2000:2005.
hourly	Should hourly means be calculated? The default is TRUE. If FALSE then the raw data are returned.
n.cores	Number of cores to use for parallel processing. Default is 1 and hence no parallelism.
quiet	If FALSE, print missing sites / years to the screen.
path	If a file path is provided, the data are saved as an rds file at the chosen location e.g. path = "C:/Users/David". Files are saved by year and site.

## Details

Note the following units for the main variables:

**date** Date/time in POSIXct format. **Note the time zone is GMT (UTC) and may need to be adjusted to merge with other local data. See details below.**

**latitude** Latitude in decimal degrees (-90 to 90).

**longitude** Longitude in decimal degrees (-180 to 180). Negative numbers are west of the Greenwich Meridian.

**elevation** Elevation of site in metres.

**wd** Wind direction in degrees. 90 is from the east.

**ws** Wind speed in m/s.

**ceil\_hgt** The height above ground level (AGL) of the lowest cloud or obscuring phenomena layer aloft with 5/8 or more summation total sky cover, which may be predominantly opaque, or the vertical visibility into a surface-based obstruction.

**visibility** The visibility in metres.

**air\_temp** Air temperature in degrees Celcius.

**dew\_point** The dew point temperature in degrees Celcius.

**atmos\_pres** The sea level pressure in millibars.

**RH** The relative humidity (%).

**cl\_1, ..., cl\_3** Cloud cover for different layers in Oktas (1-8).

**cl** Maximum of cl\_1 to cl\_3 cloud cover in Oktas (1-8).

**cl\_1\_height, ..., cl\_3\_height** Height of the cloud base for each later in metres.

**precip\_12** 12-hour precipitation in mm. The sum of this column should give the annual precipitation.

**precip\_6** 6-hour precipitation in mm.

**precip** This value of precipitation spreads the 12-hour total across the previous 12 hours.

**pwc** The description of the present weather description (if available).

The data are returned in GMT (UTC). It may be necessary to adjust the time zone when combining with other data. For example, if air quality data were available for Beijing with time zone set to "Etc/GMT-8" (note the negative offset even though Beijing is ahead of GMT. See the `openair` package and manual for more details), then the time zone of the met data can be changed to be the same. One way of doing this would be `attr(met$date, "tzone") <- "Etc/GMT-8"` for a meteorological data frame called `met`. The two data sets could then be merged based on date.

## Value

Returns a data frame of surface observations. The data frame is consistent for use with the `openair` package. Note that the data are returned in GMT (UTC) time zone format. Users may wish to express the data in other time zones, e.g., to merge with air pollution data. The `lubridate` package is useful in this respect.

**Author(s)**

David Carslaw

**See Also**[getMeta\(\)](#) to obtain the codes based on various site search approaches.**Examples**

```
## Not run:  
## use Beijing airport code (see getMeta example)  
dat <- importNOAA(code = "545110-99999", year = 2010:2011)  
  
## End(Not run)
```

---

weatherCodes

*Codes for weather types*

---

**Description**

This data frame consists of the weather description codes used in the ISD. It is not of general use to most users.

**Examples**

```
## basic structure  
head(weatherCodes)
```

# Index

## \* datasets

weatherCodes, 7

exportADMS, 2

getMeta, 3

getMeta(), 7

getMetaLive, 4

getMetaLive(), 4

importNOAA, 5

importNOAA(), 2, 4

lubridate, 6

tibble, 5

weatherCodes, 7